Abbott

R/Pharma

Nov | 2025

Generating Synthetic Data with *synthpop* in R

Sophie Furlow

Statistician, Abbott Diagnostics Division

sophie.furlow@abbott.com

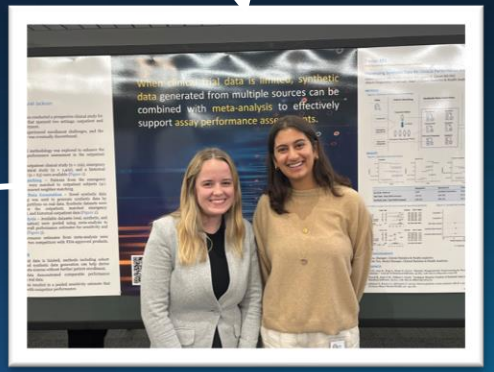
Abbott

R/Pharma

Nov | 2025

B.S. Bioengineering
Northwestern University

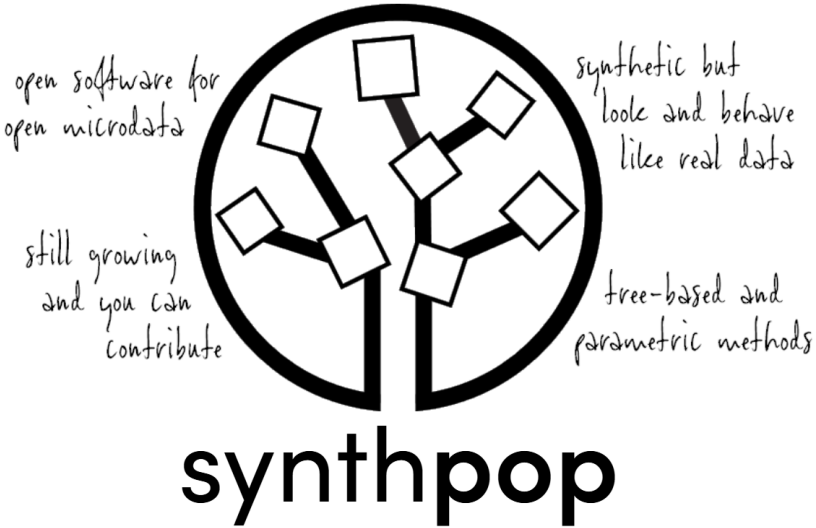
M.Eng. Bioengineering
UC Berkeley



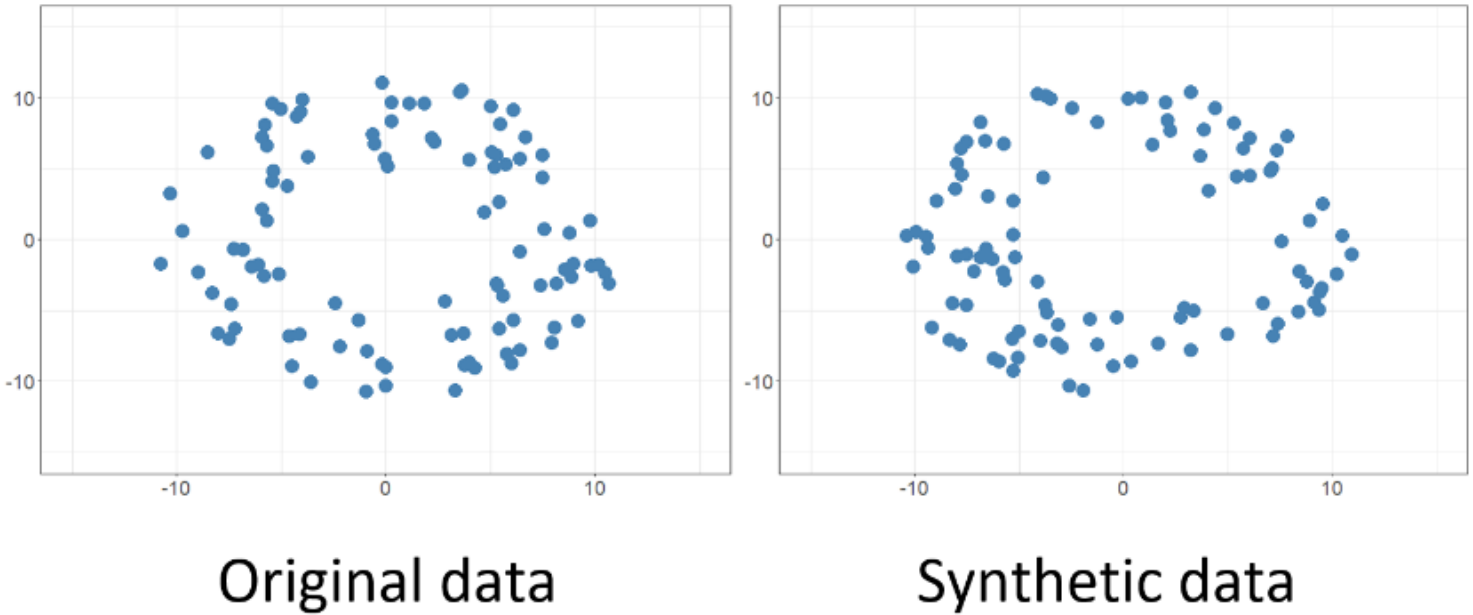
Statistician
Abbott

Agenda

1. Introduction to synthetic data
2. Basics of *synthpop*
3. Utility & privacy evaluation
4. Application: model training



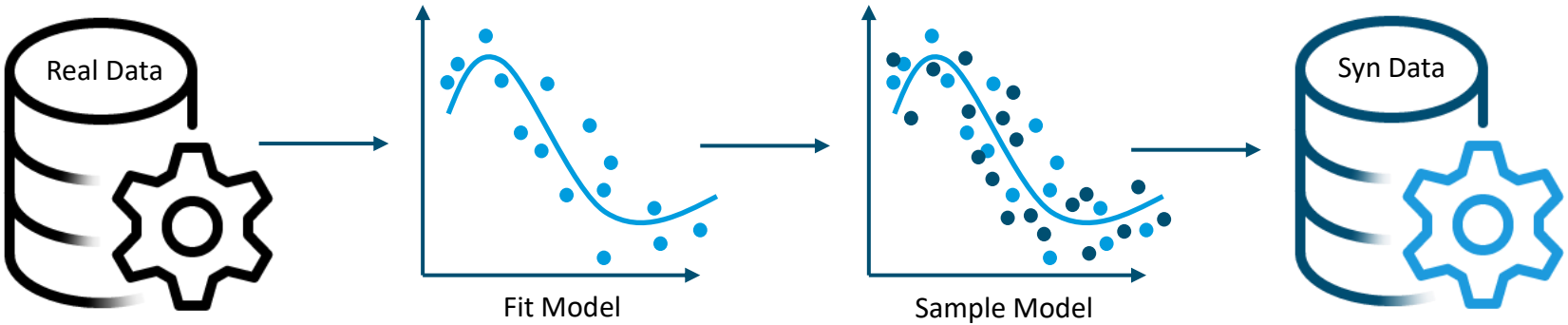
Synthetic data is a new set of records that mimics a real dataset



https://www.geeksforgeeks.org/data-science/what-is-synthetic-data/

Synthetic data is generated by applying algorithms to real data

- Generate from RWD, clinical trials, other health data sources
- Holy grail: synthetic data with the same *utility* as original data, but completely *anonymous*



Journal of Statistical Software

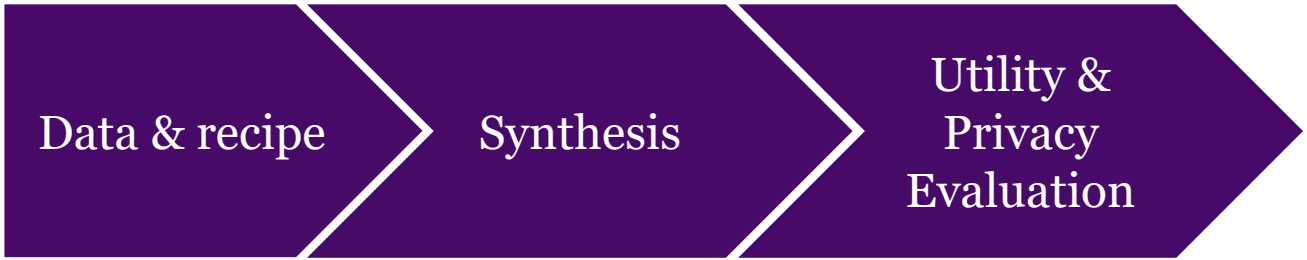
October 2016, Volume 74, Issue 11. *doi: 10.18637/jss.v074.i11*

synthpop: Bespoke Creation of Synthetic Data in R

Beata Nowok
University of Edinburgh

Gillian M. Raab
University of Edinburgh

Chris Dibben
University of Edinburgh



We will use open-source data for this demo

- National Health And Nutrition Examination Survey (NHANES)
- National Center for Health Statistics (NCHS) mortality data
- Our training set
  - 11,282 subjects (4,834 in test)
  - Participants in NHANES 1999-2004

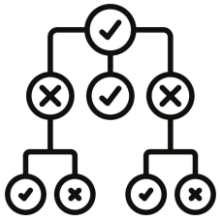
https://www.cdc.gov/nchs/data-linkage/mortality-public.htm
https://www.cdc.gov/nchs/nhanes/about/

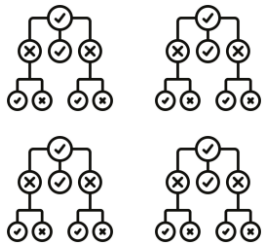
`codebook.syn(real_train)`

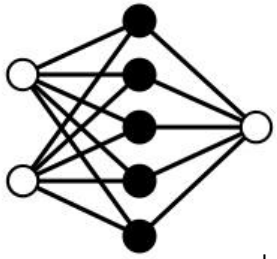
variable	class	nmiss	perctmiss	ndistinct	details
age	integer	0	0.00	68	
bmi	numeric	1176	10.42	2514	
ethnicity	factor	0	0.00	5	
education	factor	1160	10.28	7	
marital	factor	341	3.02	8	
income	factor	1223	10.84	15	
sex	factor	0	0.00	2	
smoking	factor	0	0.00	2	
diabetes	factor	9	0.08	3	
hypertension	factor	0	0.00	2	
ntprobnp	numeric	2166	19.20	6161	
egfr	factor	1465	12.99	2	
mortality	factor	0	0.00	2	
cod	factor	7984	70.77	10	

Methods for *synthpop* recipe

CART	Parametric	Random Forest	Neural Network
Classification and regression trees	*synthpop's* choice per variable: NormRank/LogReg /PolyReg/POLR	Ensemble method using CARTs as building blocks	Simple one-layer neural network classifier







Generation of 3 synthetic copies

```
syn(data = real_train, method = method, m = 3)
```

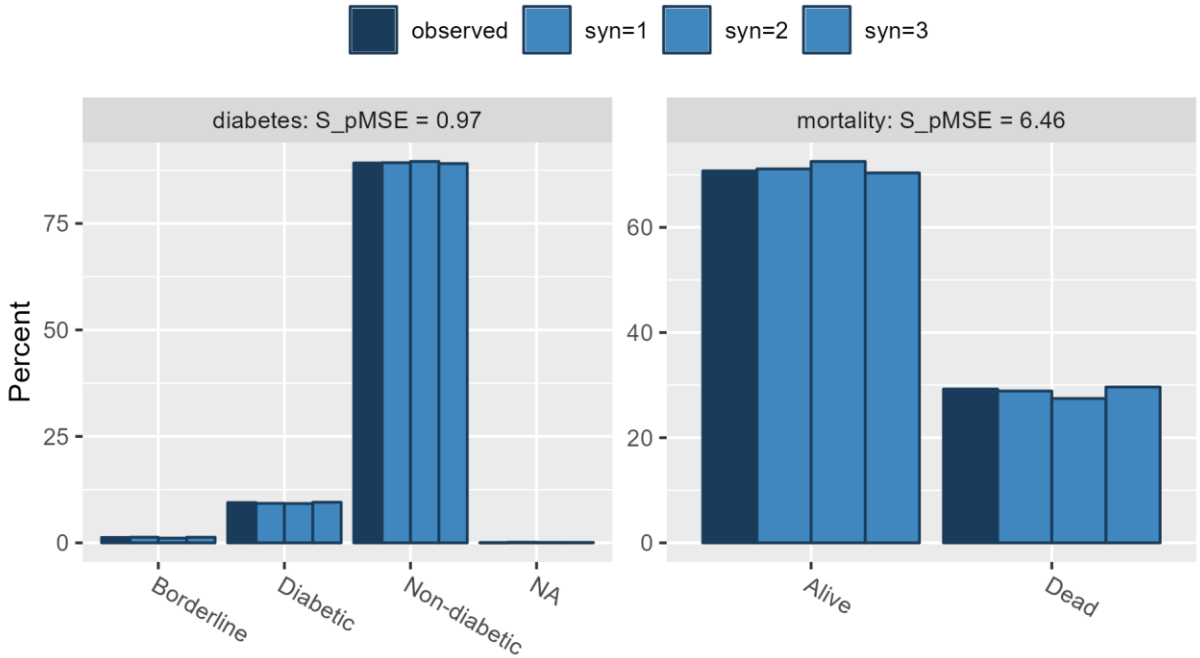
age	bmi	ethnicity	education	marital	income
38	22.69	Mexican American	Some College or AA Degree	Married	*NA*
61	27.26	Other Hispanic	Less Than 9th Grade	Married	$65000-74999
26	*NA*	Other Hispanic	High School Grad/GED or Equivalent	Never Married	$75000 and Ove
73	23.99	Non-Hispanic Black	Some College or AA Degree	Married	*NA*
20	52.81	Non-Hispanic Black	High School Grad/GED or Equivalent	Never Married	$20000-24999
18	28.13	Mexican American	*NA*	Never Married	$25000-34999

sex	smoking	diabetes	hypertension	ntprobnp	egfr	mortality	cod
Female	Non-smoker	Non-diabetic	No hypertension	65.82	Normal	Alive	*NA*
Female	Non-smoker	Diabetic	No hypertension	14.05	Normal	Alive	*NA*
Male	Smoker	Non-diabetic	No hypertension	31.18	Normal	Alive	*NA*
Female	Smoker	Diabetic	Hypertension	*NA*	<60	Dead	Cardiovascular
Female	Smoker	Non-diabetic	No hypertension	*NA*	*NA*	Alive	*NA*
Female	Smoker	Non-diabetic	No hypertension	25.75	Normal	Alive	*NA*

Utility:
How similar is my synthetic data to the real data?

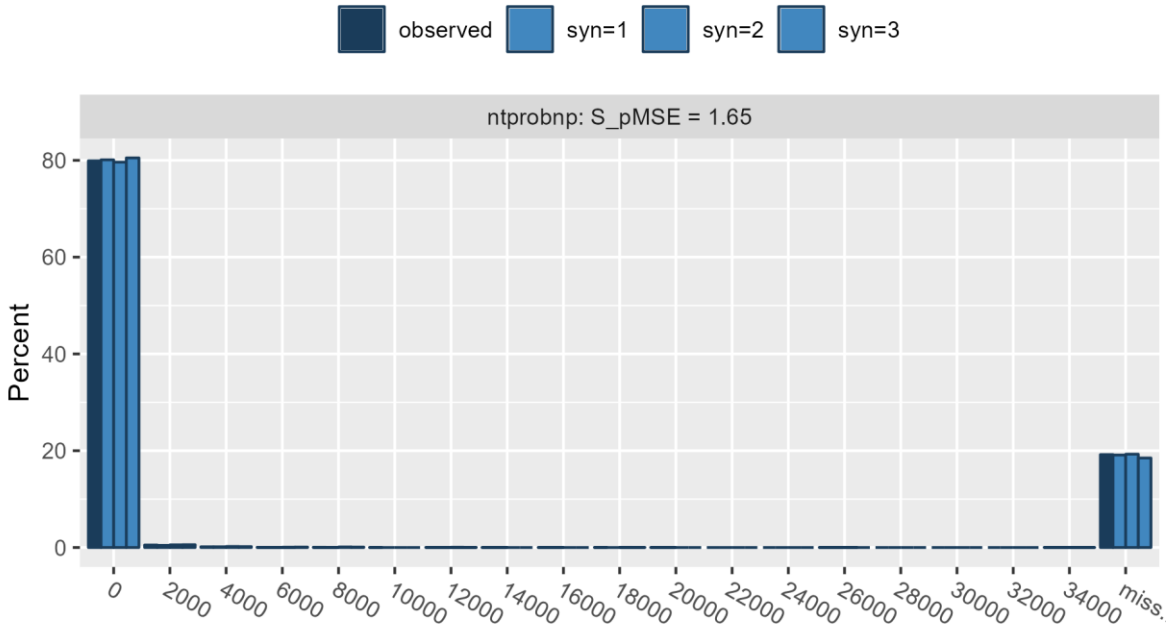
Univariate distributions of synthetic and original variables are comparable

```
cmp_cart <- compare(syn_cart, real_train, vars = c('diabetes', 'mortality', 'ntprobnp'))
```



Univariate distributions of synthetic and original variables are comparable

```
cmp_cart <- compare(syn_cart, real_train, vars = c('diabetes', 'mortality', 'ntprobnp'))
```



Global utility measures

- Propensity Mean Squared Error (pMSE)
  - Measures distinguishability of synthetic data from real
  - Calculated by creating a classifier to label real and synthetic records in a combined dataset: pMSE is MSE between model's predicted probabilities and true sources of each record
- Standardized pMSE (S_pMSE)

Low pMSE and S_pMSE values are desirable

Global utility is highest for CART

```
util_cart <- utility.gen(syn_cart, data = real_train)
```

	Mean pMSE	Mean S_pMSE
CART	0.00454	2.89
NeuralNet	0.0108	7.06
Parametric	0.0110	7.02
RandomForest	0.0170	10.50

Privacy: How susceptible to re-identification is our synthetic data?

All 4 generative models reduce disclosure risk

Comparison of attribute disclosure measures: DiSCO for synthetic data to Dorig for original data.

Dorig

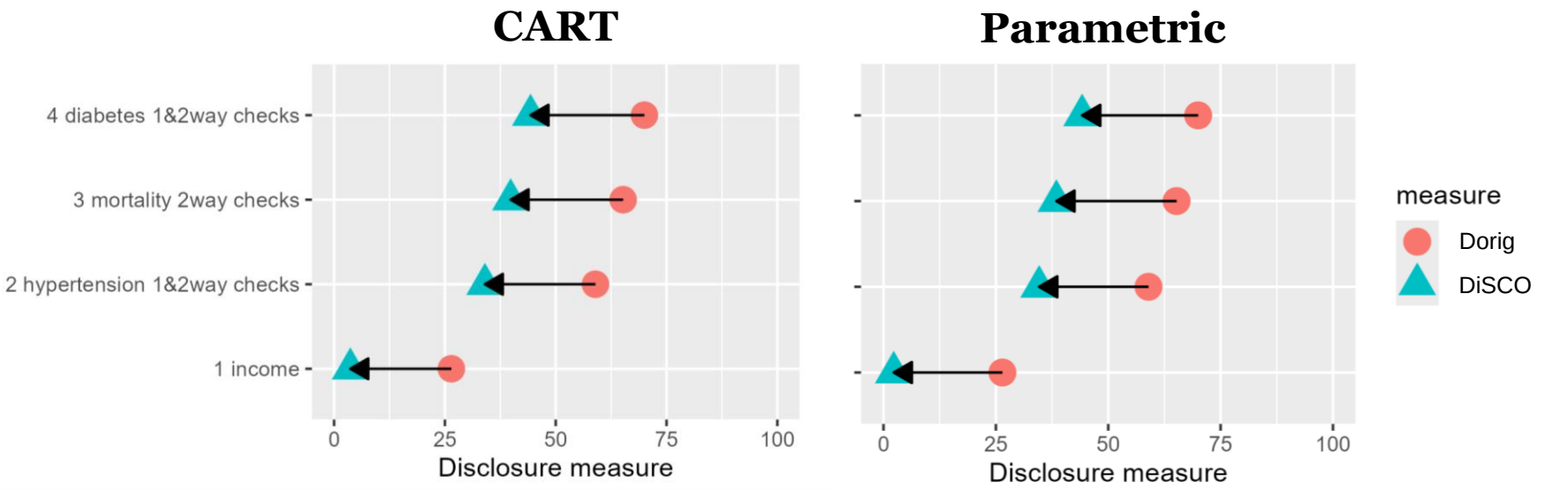
- Measures diversity of the original dataset – linkage of predictor values to unique target values

DiSCO

- Measure of diversity in the synthetic dataset – how many combos of variable values link to a unique and correct target value?

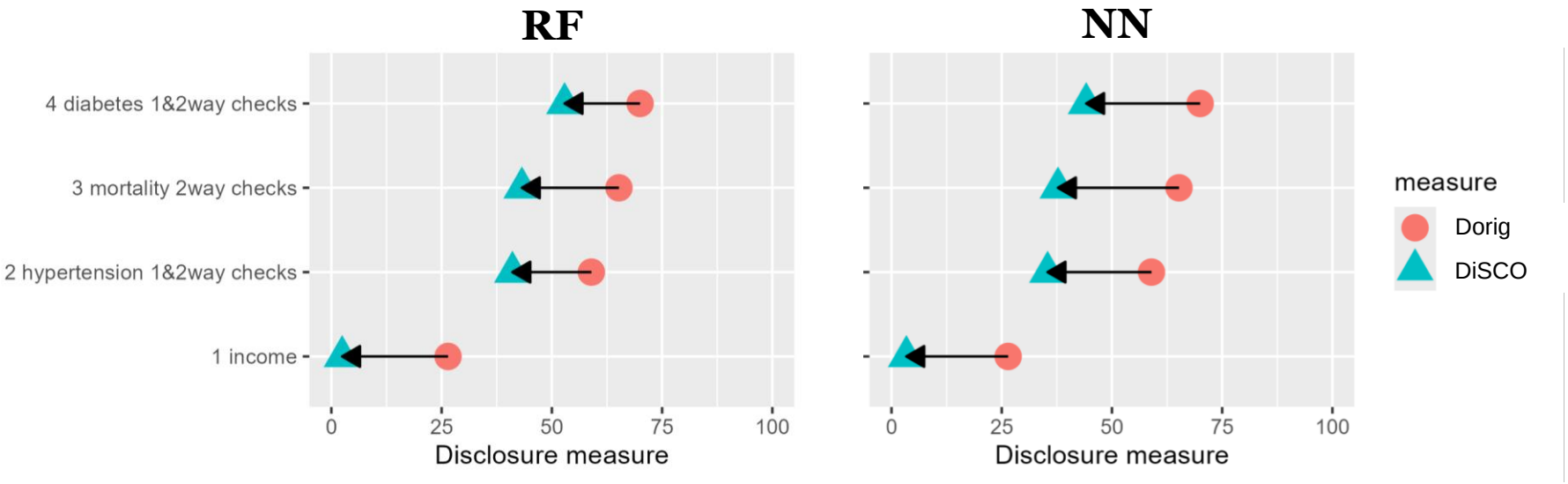
All 4 generative models reduce disclosure risk

```
disc_cart <- multi.disclosure(syn_cart, real_train, keys, targets)
```



All 4 generative models reduce disclosure risk

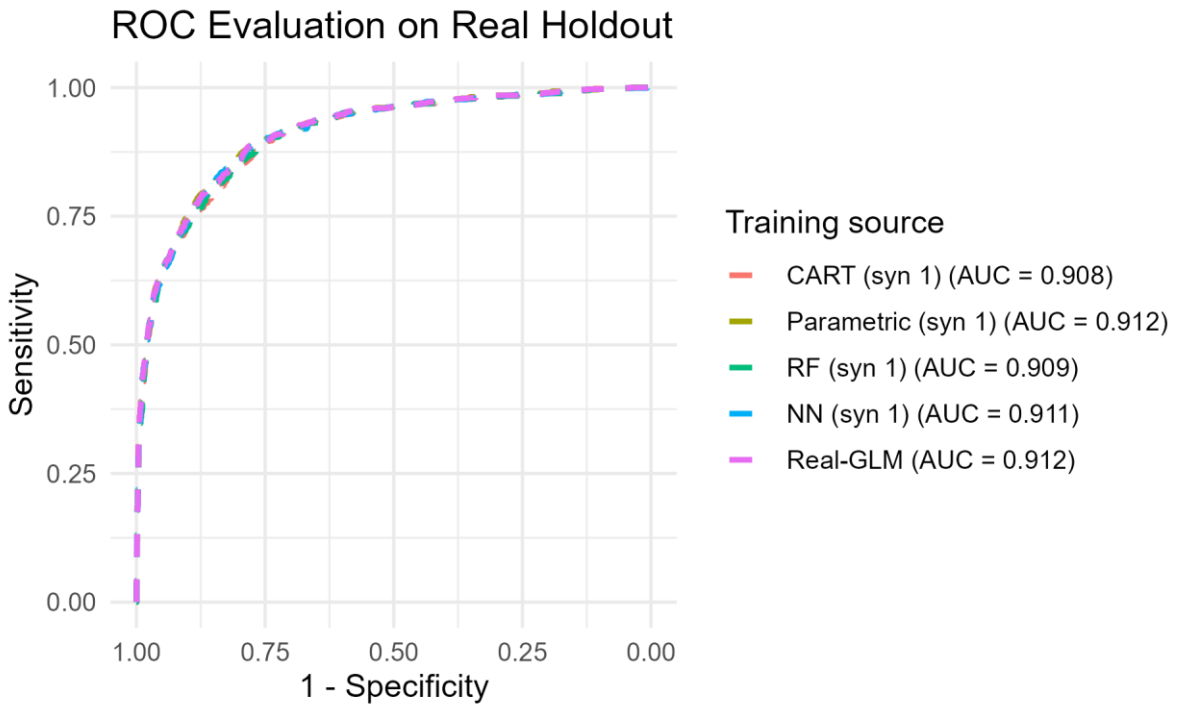
```
disc_cart <- multi.disclosure(syn_cart, real_train, keys, targets)
```



Application:
Can synthetic datasets train machine learning models?

LR models trained on synthetic data perform well on real holdout set

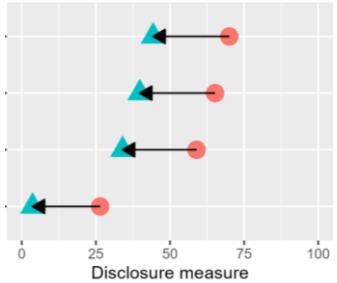
- Train logistic regression models to classify mortality on each synthetic dataset
- Predict mortality outcome for each record in real holdout set

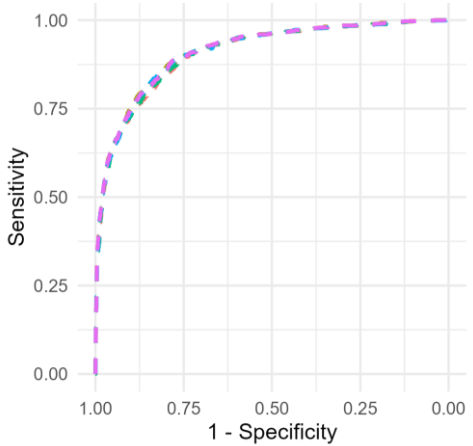


Conclusions

- All 4 generative methods reduced disclosure risk while preserving global utility of the original dataset
- Logistic regression models trained on all 4 synthetic sets can perform well on a real hold out set

	Mean pMSE	Mean S_pMSE
CART	0.00454	2.89
NeuralNet	0.0108	7.06
Parametric	0.0110	7.02
RandomForest	0.0170	10.50





Abbott

THANK YOU!

For questions and code access:

sophie.furlow@abbott.com

sfurlow8

Abbott

Appendix

Synthpop recipe

Visit Sequence	Prediction Matrix	Method	Strata
Order in which to synthesize the input variables	Which vars predict other vars?	Generative models: CART, parametric, random forest, neural network	Synthesize by group of specified variables

Synthpop recipe: visit sequence

```
# visit sequence: synthesize predictors first, outcome last
visit_seq <- c(
  'age',
  'sex',
  'ethnicity',
  'bmi',
  'education',
  'marital',
  'income',
  'smoking',
  'diabetes',
  'hypertension',
  'ntprobnp',
  'egfr',
  'mortality',
  'cod'
)
```

Synthpop recipe: methods

```
# cart
method_cart <- 'cart'

# parametric (lets synthpop choose defaults per type: normrank/logreg/polyreg/polr)
method_parametric <- "parametric"

# random forest
method_rf <- 'rf'

# neural network
method_nn <- rep("cart", ncol(real_train))
names(method_nn) <- names(real_train)
method_nn[c('bmi', 'smoking', 'diabetes', 'mortality')] <- 'nn'
```

Synthpop recipe: prediction matrix

```
# predictor matrix: use all-others as predictors
pred_mat <- matrix(
  1,
  ncol = ncol(real_train),
  nrow = ncol(real_train),
  dimnames = list(names(real_train), names(real_train))
)
diag(pred_mat) <- 0 # variable does not predict itself
```

	age	bmi	ethnicity	education	marital	income	sex
age	0	0	0	0	0	0	0
bmi	1	0	1	0	0	0	1
ethnicity	1	0	0	0	0	0	1
education	1	1	1	0	0	0	1
marital	1	1	1	1	0	0	1
income	1	1	1	1	1	0	1
sex	1	0	0	0	0	0	0